A Constrained Multi-objective Evolutionary Strategy based on Population State Detection

Huanrong Tang\textsuperscript{a,b}, Fan Yu\textsuperscript{1a,b}, Juan Zou\textsuperscript{a,b,*}, Shengxiang Yang\textsuperscript{a,d}, Jinhua Zheng\textsuperscript{a,b,c}

\textsuperscript{a}Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Computer science College of Xiangtan University, Xiangtan, Hunan Province, China
\textsuperscript{b}School of Computer Science and School of Cyberspace Science University of Xiangtan, Xiangtan, 411105, China
\textsuperscript{c}Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China
\textsuperscript{d}School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.

Abstract

The difficulty of solving constrained multi-objective optimization problems (CMOPs) using evolutionary algorithms is to balance constraint satisfaction and objective optimization while fully considering the diversity of the solution set. Many CMOPs with disconnected feasible subregions make it difficult for algorithms to search for all feasible nondominated solutions. To address these issues, we propose a population state detection strategy (PSDS) and a restart scheme to determine whether the environmental selection strategy needs to be changed based on the situation of population. When the population converges in the feasible region, the unconstrained environmental selection allows the population to cross the current feasible region. When the population converging outside the feasible region, all constraints will be considered in the environmental selection to select the population for the feasible region. In addition, the restart scheme will use reinitialization to make the population jump out of unprofitable iterations. The proposed algorithm enhances the search ability through the detection strategy and provides more diversity by reinitializing the population. The experimental results on four constraint test suites with various features have demonstrated that the proposed algorithm had better or competitive performance against other state-of-the-art constrained multi-objective algorithms.

Keywords: Constrained Multi-objective optimization, evolutionary algorithm, state detection, constraint handling, restart scheme.

1. INTRODUCTION

Constrained multi-objective optimization problems (CMOPs) widely exist in real-world production and practical engineering \cite{1,2,3}. CMOPs usually involve a number

\textsuperscript{1}Co-first author: Huanrong Tang, Fan Yu (These authors have contributed equally to this work.)
\textsuperscript{*}Corresponding author: Juan Zou

Email address: zoujuan@xtu.edu.cn (Juan Zou)
of conflicting objectives simultaneously with a series of constraints. Without loss of
gen[46]erality, a CMOP can be defined as follows [4]:

\[
\begin{align*}
\text{minimize} & \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\
\text{subject to} & \quad g_i(\mathbf{x}) \geq 0, \quad i = 1, \cdots, p \\
& \quad h_j(\mathbf{x}) = 0, \quad j = p + 1, \cdots, p + q \\
& \quad \mathbf{x} \in \mathbb{R}^n
\end{align*}
\]

where \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) is an \( n \)-dimensional decision variable vector from the decision
space \( \mathbb{R}^n \); \( \mathbb{R}^m \) is the objective space. \( \mathbb{R}^m \to \mathbb{R}^n \) is an objective function vector that consists of \( m \) conflicting objective functions, and \( \mathbf{F}(\mathbf{x}) \in \mathbb{R}^n \). \( g_i(\mathbf{x}) \geq 0 \) is the \( i \)th inequality con-
straints, \( p \) is the number of inequality constraints, \( h_j(\mathbf{x}) = 0 \) is the \( j \)th equality constraint, \( q \) is the number of equality constraints.

To solve CMOPs with many inequality constraints and possible equality constraints, this work quantifies the constraint satisfaction degree of \( x \) on each constraint function. Thus, the constraint violation of \( \mathbf{x} \) on the \( j \)th constraint can be defined as follows:

\[
C_j(\mathbf{x}) = \begin{cases} 
\max(0, g_j(\mathbf{x})), & j = 1, \ldots, p \\
\max(0, |h_j(\mathbf{x})| - \delta), & j = p + 1, \ldots, p + q 
\end{cases}
\]

where \( \delta \) is an introduced extremely small positive number (e.g., \( \delta = 10^{-6} \)) to convert equality constraints into inequality constraints [3]. The overall constraint violation (CV) can be used in various CMOPs to evaluate the degree of constraint violation of \( \mathbf{x} \), which can be summarized as:

\[
CV(\mathbf{x}) = \sum_{j=1}^{p+q} C_j(\mathbf{x}).
\]

If there is a solution \( \mathbf{x} \in \mathbb{R}^n \) that does not violate any constraint function, then it is called a feasible solution and its overall constraint violation satisfies \( CV(\mathbf{x}) = 0 \); otherwise, \( \mathbf{x} \) is called an infeasible solution, and the larger the value of \( CV(\mathbf{x}) \), the higher the degree of CV by \( \mathbf{x} \). The solution space that satisfies all constraints is called the feasible region; otherwise, it is called the infeasible region. The feasible solution set consisting of all feasible solutions can be defined as:

\[
S = \{ \mathbf{x} \mid CV(\mathbf{x}) = 0, \mathbf{x} \in \mathbb{R}^n \}.
\]

For any two solutions \( \mathbf{x}_a, \mathbf{x}_b \in S \). Since both of them are feasible solutions, it is important
to distinguish which is better in objective functions. \( \mathbf{x}_a \) is said to dominate \( \mathbf{x}_b \) if \( f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b) \) for each \( i \in \{1, \ldots, m\} \) and \( f_j(\mathbf{x}_a) < f_j(\mathbf{x}_b) \) for at least one \( j \in \{1, \ldots, m\} \), denoted as \( \mathbf{x}_a \prec \mathbf{x}_b \). If no other solution from \( S \) can dominate \( \mathbf{x}^* \), then \( \mathbf{x}^* \) is called a feasible nondominated solution or Pareto optimal feasible solution. The set of all feasible
nondominated solutions is called the Pareto optimal set (PS). The mapping of the feasible PS in the objective space is called the constrained Pareto Front (PF). The primary
process to solve CMOPs is to allow the population to converge to the constraints PF and spread along the feasible boundaries in every feasible nondominated region [3].

Multi-objective evolutionary algorithms (MOEAs) are effective and extremely reliable in solving multi-objective optimization problems (MOPs) [3, 4, 5]. They can provide
a set of nondominated approximate optimal solutions for a series of conflicting objectives. The characteristics of MOEAs are robust and adaptive to cope with increasingly complex MOPs, such as CMOPs and dynamic multi-objective optimization problems (DMOPs) [10], [11]. MOEAs can be roughly divided into three categories [12]. The first category includes the algorithms based on Pareto dominance that enhance population convergence, such as NSGAII [13]. The second category is based on decomposition that can predefine a set of uniformly distributed weight vectors to ensure population diversity, such as MOEA/D [14]. The third category is based on indicators that can evaluate the convergence and diversity of a population simultaneously, such as IBEA [15] and SMS-EMOA [16]. These common MOEAs, which are mainly designed to solve MOPs, cannot handle CMOPs well. Therefore, constraint handling mechanisms in evolutionary computation are designed to solve CMOPs better [17].

The vital challenge for constrained multi-objective evolutionary algorithms (CMOEAs) to solve CMOPs is to maintain a better balance between the optimization of objectives and constraint satisfaction to achieve better convergence [18]. Among the algorithms for solving CMOPs, the first and most representative category is the constraint dominance principle (CDP) algorithm. This type of algorithm always chooses feasible solutions that are better than infeasible solutions. Because the population is blocked by large infeasible regions [19], they cannot optimize the convergence on the objective function. The second category of CMOEAs treats infeasible solutions with fewer constraint violations as feasible solutions. This type of algorithm improves search ability in infeasible regions but weakens the search for tiny feasible regions [20] as well as convergence in the feasible regions.

This paper proposes an evolutionary strategy to detect the state of the population and provide feedback to the next environmental selection during the evolution process. To be more specific, the main contributions of this work can be summarized as follows:

1) The proposed population state detection strategy (PSDS) can detect whether the population converges to the boundary of the feasible region or the unconstrained PF outside the feasible region. Based on the results of the population status detection, the algorithm decides whether to consider constraints in the environment selection. This strategy can maximize the convergence ability in all feasible regions and the search ability in infeasible regions.

2) A restart scheme by reinitializing the population lets the population end the unprofitable iteration. Additionally, an archive is set to store and update the feasible nondominated solutions in each generation. This archive set is used to make up for losing excellent solutions caused by reinitialization, guaranteeing that changing the environmental selection strategy and reinitializing the population can always benefit.

The remainder of this paper is organized as follows. Section 2 introduces the existing MOEAs for CMOPs and the motivation of the work. Section 3 illustrates the details of the algorithm. Experimental results from comparing the proposed PSDS with other CMOEAs are detailed in Section 4. Finally, conclusions are presented in Section 5.
2. RELATED WORK

In this section, the existing MOEAs with Constraint Handling Technique (CHT) are introduced. We also classify these CMOEAs and analyze their strengths and weaknesses, as well as describe our motivations.

2.1. Existing CMOEAs

Using CMOEAs to solve CMOPs requires that all Pareto optimal solutions obtained by the algorithm on the objective functions satisfy all constraints simultaneously. The early MOEAs treated the objective functions and the constraint functions equally. For instance, there exist a common CHT in the early CMOEAs which applies the violation of each constraint to every solution to weaken the performance of infeasible solutions when solving constraint problems [21]. These are collectively called penalty-based CHTs. The penalty function method uses the penalty factors to convert a CMOP into an unconstrained optimization problem. When the penalty factors are too small, many infeasible solutions will have a competitive advantage, which will cause the population to gather in feasible regions. When the penalty factors are too large, the advantage of solutions with better objective values is greatly weakened, which causes the feasible solutions to converge poorly to the PF. Although such CHTs are straightforward and effective for some simple constraints, it is difficult to quickly match the most suitable penalty factors to make a good balance between various constraints and objectives [22].

Since the feasibility of solutions is fundamental in CMOPs, the most representative MOEAs use the CDP as the main selection mechanism [13]. This means that when determining the dominance relation, satisfying constraints have a higher priority than optimizing objectives. For example, in NSGAII, the principle of feasibility is embedded in the fast nondominated sorting, which means that feasible solutions always dominate infeasible solutions. Otherwise, an infeasible solution with low constraint violations also dominates solutions with high constraint violations. If a solution \( x \) is defined to dominate another solution \( y \), then they must satisfy the following definition:

\[
x \preceq_{\text{CDP}} y \iff \begin{cases} 
g(x) < g(y), & \text{if } CV(x) = CV(y) = 0 \\
CV(x) < CV(y), & \text{otherwise.}
\end{cases}
\]

The idea of prioritizing feasible solutions is also widely used in decomposition-based MOEAs [23]. It is the priority of these algorithms to select the individual with the lowest \( CV \) when updating the solution of a weight vector. When there are multiple feasible individuals, their aggregation function values are used as the second evaluation criteria, and the best one is retained on the weight vector. As CMOPs become more complex, existing CMOEAs that only select feasible solutions uniformly converge in the first feasible region found due to excessive selection pressure. They must obtain as much beneficial information as possible from the infeasible solutions in order to solve complex CMOPS, such as those that have discontinuous feasible regions. There are many algorithms that treat infeasible solutions with minor CVs as feasible solutions, thus relaxing the constraints on feasibility. An example is the ACDP [24], which defines an angle between two infeasible solutions that exceed a threshold. When this included angle exceeds the threshold, these two solutions are considered nondominated by each other. If a random decimal in the range of (0, 1) is less than the feasible solution ratio (fr)
of the current population, one of the nondominated solutions that has a lower objective function value is chosen. The dominance of the two solutions $x$ and $y$ in ACDP can be defined as follows:

$$x \preceq_{ACDP} y \iff \begin{cases} g(x) < g(y), & \text{if } CV(x) = CV(y) = 0 \\ CV(x) < CV(y), & \text{else if } \text{angle}(x, y) \leq \theta \\ g(x) < g(y), & \text{else if } \text{rand}(0, 1) < fr \end{cases}$$

where $\text{angle}(x, y)$ is the angle between $x$ and $y$; the parameter $\theta$ is the angle threshold, and $fr$ represents the ratio of feasible solutions in the current population. Another idea, called the $\varepsilon$ Constraint method (EC) \[^25\], sees an infeasible solution whose constraint violation is less than $\varepsilon$ as a feasible solution. The introduced constraint permission value can control the search ability of the population in infeasible regions, and the value of $\varepsilon$ usually changes dynamically according to the needs of the population’s evolution process.

The order of two solutions $x$ and $y$ in EC is defined as follows:

$$x \preceq_{EC} y \iff \begin{cases} g(x) < g(y), & \text{if } CV(x) \leq \varepsilon \text{ and } CV(y) \leq \varepsilon \\ \text{or } CV(x) = CV(y) \\ CV(x) < CV(y), & \text{otherwise} \end{cases}$$

where $x \preceq_{EC} y$ means that $x$ is better than $y$ when the constraint permission value is $\varepsilon$, which is adaptively controlled by the mean value of the overall constraint violations $CV_{\text{mean}}$ and the $fr$ \[^25\] are as follows:

$$\varepsilon = CV_{\text{mean}} \times fr.$$  

The search progress between feasible regions and infeasible regions and the performance on different DD CMOPs are primarily affected by the dynamic adjustment mechanism of $\varepsilon$.

There are remarkable benefits brought by the exploration of infeasible areas when the EC solves CMOPs. Therefore, there exist some $\varepsilon$ constraint-handling methods like (MOEAD-I-Epsilon) \[^26\] that can achieve better results. In addition to this type of $\varepsilon$ constraint-handling mechanism that improves the $\varepsilon$ adjustment strategy, there is also a novel selection mechanism inspired by the EC called $\varepsilon$MOEA/D-$\sigma$ \[^27\], which does not adjust the $\varepsilon$ value separately like the ordinary improved version of EC but creatively builds a bi-objective optimization problem that considers the scalarizing function and the constraint violation degree as an objective function. When the constraint violation value of the solutions is generally far away from $\varepsilon$, there is no need to consider the influence of the constraint. Using a scalar function to select the optimal solution reduces the impact of constraints on the distribution of population convergence. Another new type of solution evaluation algorithm proposed by Roy de Winter has a significant effect in solving CMOPs. This method of using Radial Basis Functions Approximation to fit goals and constraints is called SAMO-COBRA \[^28\]. This optimization method evaluates the comprehensive performance of the solution, similar to reducing the objective and constrained dimensions, unexpectedly does not produce defects. Such as inaccurate expression of the characteristics of the solution. This good performance is mainly due to the proper and effective use of hyper-parameter tuning, which can automatically determine the best radial basis function to achieve the best fitting effect.
At the same time, many CMOEAs with multiple stages of evolutionary progress have been proposed to adjust the strategies used for the different situations for solving CMOPs. The ToP \([\text{29}]\) proposes a two-stage evolutionary framework. In the first stage, all constraints and a single objective are considered for optimization. Then all objectives and constraints in the second stage are considered for optimization. The ToR \([\text{30}]\) sets the fitness function for evaluating each solution as the weighted sum of two ranks. CDP ranks one, another is ranked by Pareto dominance. The proportion of the two ranks is dynamically adjusted as the iteration progresses to search the objective space better. In the Push-and-Pull Strategy (PPS) \([\text{31}]\), the evolution process is divided into a push stage and a pull stage. The population in PPS evolves by ignoring any constraints and then quickly converges to the unconstrained PF. When the population has converged to the unconstrained PF, the pull stage is activated to let the process of evolution consider all constraints.

In the framework of co-evolution, the two-archive evolutionary algorithm (C-TAEA) \([\text{32}]\) is a dual-population co-evolution algorithm. The convergence-oriented archive (CA) is evolved by optimizing both objectives and constraints. Another diversity-oriented archive (DA) is evolved to optimize only the objectives. two populations interact in the mating selection and environmental selection. Inspired by C-TAEA, the coevolutionary framework for constrained multi-objective optimization (CCMO) \([\text{33}]\) is also a dual-population evolutionary algorithm in which one population considers constraints and the other population ignores constraints. Compared with the C-TAEA, CCMO adopts a weak interaction strategy that allows the two populations to exchange helpful information only among their offspring. These two weakly interacting populations in CCMO can communicate with each other without adverse effects.

Nowadays, in the research of CMOPs, in addition to proposing practical algorithms or strategies, it is also of great significance to create effective and challenging constrained multi-objective test suites. There are some constraint benchmarks with unique designs and practical significance. In Eq-(I)DTLZ \([\text{34}]\), proposed by Oliver Cuate et al., the commonly used method of converting an equality constraint problem into an inequality constraint problem deviates significantly when faced with high dimensional search spaces. Therefore, a suite of CMOPs containing true equality constraints has been proposed. This benchmark is scalable both in the decision and objective space and in the number of equality constraints. Another study is a new framework for constrained test problem construction proposed by Yuren Zhou and Yi Xiang \([\text{35}]\). This framework uses one type of convergence-hardness to introduce infeasible barriers when the population is approaching the optima, another kind of diversity-hardness to restrict the feasible optimal regions to obtain different shapes of PF. The scalable and constrained suite designed based on this framework covers a variety of difficulties.

2.2. Motivation of this work

The key to solving CMOPs is to balance objective optimization and constraint satisfaction. However, many existing CMOPs may have feasible regions separated by large infeasible regions \([\text{19}]\) or their feasible regions are discrete and far apart from each other \([\text{20}]\), which causes most CMOEAs to encounter various difficulties in searching these complex feasible regions. The feasible region and constrained PF corresponding to two common types of complex CMOPs are shown in Fig. 1. In Fig. 1(a), since the infeasible
region blocks feasible regions, the convergence direction is discontinuous in the convergence direction, which means that there exists a local optimal feasible region dominated by other feasible regions. In Fig. 1 (b), there are multiple nondominated sub-feasible regions, and even some sub-feasible regions are very narrow and small, which will cause the solution generated by the algorithm to inadvertently skip a small feasible region and be difficult to find again.

To illustrate the dilemmas more intuitively, Fig. 2 plots the populations obtained by the three most representative algorithms NSGAII-CDP [13], PPS [31] and C-TAEA [32] on C1-DLTZ3 [19] and MW13 [20], respectively. The features of C1-DTLZ3 and MW13 meet the two types of dilemmas mentioned. From the results of the algorithm running on CMOPs in Fig. 2, it can be seen that the existing excellent algorithms still cannot solve some complex CMOPs well. This is precise because the algorithms cannot better balance objective optimization and constraint satisfaction, resulting in the solutions not being able to converge uniformly on all PFs. This paper proposes an evolution strategy based on population state detection, which can analyze the current state of the population in the evolution process through the detected parameters to solve the problems caused by various complex situations effectively. Additionally, there is a restart schema that uses reinitialization of population individuals to help the evolutionary process escape from the insurmountable dilemma and provide more diversity for the subsequent optimization process. Therefore, setting up an archive set that is updated to save and update the feasible nondominated solutions of each generation is powerful and essential for the stability of the entire algorithm. The significant effect of the proposed PSDS on these two CMOPs is shown at the far right of Fig. 2.

3. PROPOSED ALGORITHM

3.1. Framework of PSDS

The procedure of the proposed CMOEA with a PSDS is illustrated in Fig. 3. The proposed algorithm first generates a randomly initialized population, then before meeting
Figure 2: Plot of the population points of existing advanced algorithms (NSGAII-CDP, PPS, C-TAEA) when solving complex CMOPs like C1-DTLZ3 and MW13. The two results on the far right show the optimization effect of our algorithm PSDS on these two problems.

Figure 3: The flow chart of the PSDS.
the termination conditions, iteratively carries out the fitness evaluation and environ-
mental selection to determine the parents and generate new individuals through genetic
manipulation. The effectiveness of the PSDS is mainly reflected in the detection and
judgment of the population state, which can help to automatically adjust fitness evalua-
tion strategies and environmental selection strategies according to the population state
of each generation to balance the objectives and constraints. In detail, if the \( fr \)
of the population and the proportion of nondominated (PND) individuals in the population are
close to one (it is usually set to be greater than 0.99 in PSDS), then the fitness evaluation
function will not consider constraints to allow the population to cross out of the current
feasible region and search for other better feasible regions in the direction of the uncon-
strained PF. After executing an unconstrained evolution strategy, \( PND \) approaches one
and \( fr \) approaches zero (it is usually set to be less than \( 10^{-2} \) in this algorithm), then the
population has converged to the unconstrained PF outside the feasible region. Therefore,
to ensure the overall feasibility of the population, all constraints must be considered in
the evolution strategy. In addition to the two population states that can be detected,
we will detect the change rate of \( fr \) and \( PND \) before and after ten generations \([36]\). If
the rate of change of both is less than \( 10^{-2} \), the state of the population has changed
little. At this time, the population will be reinitialized to end the unprofitable iteration
and provide more beneficial information \([37]\). The cycle of detecting the rate of change
is defined as the \( T \) generations in PSDS. An archive set is used in the entire evolution
process to save and update feasible nondominated solutions permanently. In the latter
stage of the evolution process, the outstanding individuals in this archive are involved in
CDP environmental selection to obtain more helpful information.

The proposed PSDS is presented in Algorithm 1. The algorithm generates a set of
uniformly distributed reference vectors \( W \) while randomly initializing the first generation
population with a capacity of \( N \). In the iterative process, the minimum function value
of all individuals in the population on each objective function constitutes an ideal point
defined as \( Z_{\text{min}} \). Then the state of the current population \( P \) is determined according to
the \( fr \) and \( PND \) and the corresponding fitness evaluation strategy is selected. Afterward,
\( N \) parents are selected from \( P \) by binary tournament selection based on the fitness of in-
dividuals, and \( N \) offspring \( O \) are generated based on selected parents through simulated
binary crossover \([38]\) and polynomial mutation \([39]\). The environmental selection stage
chooses whether to use the constrained fitness evaluation strategy or the unconstrained
fitness evaluation strategy according to the population state. Then \( P \) and \( O \) are com-
bined into a set \( Q \) with \( 2N \) solutions, and \( N \) outstanding individuals from \( Q \) through
environmental selection are selected as the next generation of \( P \). An archive set \( A \) is set
to update and save all global nondominant feasible solutions in each generation accord-
ing to Algorithm 2. A restart schema \([37]\) and the determination of fitness evaluation
strategy are shown in Algorithm 3.

3.2. Environmental selection and elites archive of PSDS

In the process of iteration, the evolutionary algorithm mainly retains better solutions
through environmental selection strategies and uses genetic operations to generate in-
dividuals who inherit the information of the retained excellent solutions. The environ-
mental selection strategy in the PSDS in Algorithm 1 is based on the fitness evaluation
strategy in SPEA2 \([40]\). Then perform population state detection (Algorithm 3) to decide
whether to use CDP to retain feasible solutions. At first, \( S_x \), as the solution set, stores
Algorithm 1 The framework of PSDS

Input: Population size \( N \), objective number \( M \), Restart cycle \( T \), Max evolution generation \( T_{\text{max}} \), Termination criterion

Output: \( P \) (final population)

1: \( \text{isCons} \leftarrow \) Whether to consider constraints
2: \( W \leftarrow \) Uniformly distributed reference vectors
3: \( P \leftarrow \) Randomly initialize population of size \( N \)
4: \( O \leftarrow \) Offspring of \( P \)
5: \( Z_{\min} = (z_1, \ldots, z_m) \leftarrow \) Calculate the ideal point of \( P \)
6: \( t \leftarrow \) Current evolutionary generation
7: \( T_c \leftarrow \) The threshold for judging the evolution to the end stage
8: \( A \leftarrow \) Save and update the feasible nondominated solutions in \( P \)
9: \( \text{Fitness} \leftarrow \) Calculate the fitness of solutions in \( P \) by (6)
10: \( t = 1, \text{isCons} = \text{true}, T_c = 0.7 \ast T_{\text{max}}, T = T_{\text{max}}/10 \)
11: while the termination criterion is not fulfilled do
12: \( O = \text{Selection+Crossover+Mutation}(P, \text{Fitness}) \)
13: \( Q = P \setminus O; \)
14: \( P = \text{EnvironmentalSelection}(Q, W, Z_{\min}, t, \text{isCons})(\text{Algorithm 2}) \)
15: \( A = \text{ArchiveUpdate}(P, N)(\text{Algorithm 2}) \)
16: if \( t < T_c \) then
17: \[ [\text{isCons}, P, Z_{\min}] = \text{StateDetection}(P, T, Z_{\min})(\text{Algorithm 3}) \]
18: else
19: \( P = \text{EnvironmentalSelection}(Q, A, W, Z_{\min}, \text{isCons}) \)
20: end if
21: \( t = t + 1 \)
22: end while
23: Return \( P; \)

all the solutions dominating \( x \), and another solution set \( R_y \) contains all the solutions dominated by \( y \). Then, the fitness value of solution \( x \) can be defined as:

\[
\text{fitness}(x) = \sum_{y \in S_x} |R_y| + \frac{1}{\sigma_k^x + 2}, \tag{6}
\]

where the first element on the right side of the \( \text{fitness}(x) \) is the number of all the solutions dominated by the solutions who are dominate \( x \), and the rightmost element in the formula is the reciprocal of the Euclidean distance from \( x \) to its \( k \)-th nearest neighbor, where \( k \) is the sequence number of the \( k \)-th individual after the individuals around \( x \) are sorted in increasing distance. \( \sigma_k^x \) is the Euclidean distance between \( x \) and its \( k \)-th nearest neighbor, and \( \sigma_k^x \) as the denominator plus the number 2 is to make the entire rightmost formula always satisfy \( \frac{1}{\sigma_k^x + 2} < 1 \). Obviously, the smaller the fitness value of the solution \( x \), the smaller the number of other solutions that dominate \( x \), and when \( x \) is a nondominated solution, the fitness value is only the reciprocal of the Euclidean distance between \( x \) and its neighbor, which means that \( \text{fitness}(x) < 1 \). Under particular circumstances, when the selected solutions exceeding the population capacity \( N \) are all nondominated solutions, it is necessary to use the archive truncation in [40] based on Euclidean distance to exclude
Algorithm 2 Environmental selection and elites archive of PSDS

Input: The merged population $Q$, reference vector $W$, ideal point $Z_{min}$, objective number $M$, current evolutionary generation $t$, the representative parameter of the constraint $isCons$;

Output: The next population $P$;

1: if $(M == 3)$ and $(t > Tc)$ then
2:   \[F_1, F_2, \ldots, F_{c}, \ldots\] = NDsort($Q'.objs, Q'.cons$);
3:   The individuals before $F_c$ are added to $P_{t+1}^1$ by fast nondominated sorting with constraints;
4:   $N - |\cup_{j=1}^{c-1} F_j|$ individuals from $F_c$ are added into $P_{t+1}^1$ based on reference vector $W$
5: else
6:   if $isCons == true$ then
7:     Fitness = CalFitness($Q'.objs, Q'.cons$)
     Calculating the fitness value under the constraint dominance principle
8:     $P_{t+1}^1 = \{ii \in Q' \land [CV(i) = 0 \lor Fitness(i) < 1]\}$
9:   else
10:    Fitness = CalFitness($Q'.objs$)
     Calculating fitness value only considers the objective functions
11:   $P_{t+1}^1 = \{ii \in Q' \land Fitness(i) < 1\}$
12: end if
13: if $|P_{t+1}^1| < N$ then
14:   Temp = $\{ii \in R' \land Fitness(i) \geq 1\}$
15:   Add the first $N - |P_{t+1}^1|$ solutions with the smallest fitness value in Temp to $P_{t+1}^1$
16: else if $|P_{t+1}^1| > N$ then
17:   Delete $|P_{t+1}^1| - N$ solutions from $P_{t+1}^1$ by Truncation operation
18: end if
19: end if
20: $A = \{ij \in [A, P_{t+1}^1] \land CV(j) = 0 \land |A| \leq N\}$
     An archive set $A$ uses the feasible nondominated solutions in each generation $P$ to update the solutions in its own set.
21: Return $P_{t+1}^1, A$.

some of the more crowded solutions so that the nondominated solutions can obtain a good spread. When the environmental selection requires consideration of constraints, determining the dominance between two solutions gives priority to the one with the minuscule $CV$ value; otherwise, when the two solutions are both feasible solutions or their $CV$ is the same, then the fitness value is used to determine which of the two is the nondominated solution. In another case, when the environmental selection ignores the constraints, only the fitness value is used to judge the dominance relationship between two solutions, allowing the population to cross infeasible regions. In addition, to avoid the problem of losing excellent solutions when switching between constrained environmental selection and unconstrained environmental selection multiple times, we set up an archive set $A$, with a capacity not exceeding $N$, to save the updated feasible nondominated solutions in each generation. After about 70% of the total evolutionary generation in the late stage of evolution, the population state detection is stopped and the excellent
solutions in \( A \) are added to the environmental selection process. The effects of whether the environmental selection considers constraints and the consequences of setting archive set \( A \) are shown in Figure 4.

When the constrained fitness evaluation mechanism is used to make the population converge to the current feasible area (Figure 4(a)), PSDS will switch the environmental selection to the unconstrained fitness evaluation mechanism (Figure 4(b)) to make the population converge to the unconstrained PF (Figure 4(c)). Then PSDS detects that the value of the objectives of the population can no longer be optimized and the environmental selection switches to the constraint fitness evaluation mechanism (Figure 4(d)) to allow the population to converge in the optimal feasible region.

In the algorithm process of environmental selection, when faced with three-dimensional or higher-dimensional CMOP, the algorithm introduces the idea of using uniform weight vectors to guide the solution distribution in the latter stage of evolution [11].
Algorithm 3 Population state detection and restart schema

Input: The population $P$, restart cycle $T$, the ideal point $Z_{\min}$
Output: The constraint representative value $isCons$, Population $P$, Updated ideal

1: $fr \leftarrow$ The feasible solution ratio
2: $PND \leftarrow$ The proportion of nondominated solutions
3: Function $ROC(x) \leftarrow$ Change rate of 10 generations before and after $x$
4: if $fr > 0.99$ and $PND > 0.99$ then
5:     $Z_{\min} = \min(Z_{\min}, P.objs)$
6:     $isCons = false$
7: else if $fr < 10^{-2}$ and $PND > 0.99$ then
8:     $Z_{\min} = \min(Z_{\min}, P.objs, P.cons)$
9:     $isCons = true$
10: else
11:     Keep the original $Z_{\min}$ calculation method and the original $isCons$
12: end if
13: if $ROC(fr) < 10^{-1}$ and $ROC(PND) < 10^{-1}$ then
14:     $isCons = false$
15:     if mod($t, T$) == 0 and $fr < 10^{-1}$ then
16:         $P = Initialization(N)$
17:     $isCons = true$
18: end if
19: else
20:     Keep the original $P$
21: end if
22: Return $isCons$, $P$, $Z_{\min}$;
3.3. Population state detection and restart schema

In solving CMOPs, feasibility must be considered to make solutions converge in the feasible region, but ignoring constraints can make solutions converge to the true PF faster when faced with problems such as separated feasible regions in CMOP. Many existing excellent algorithms [26, 36] adjust the relationship between objective optimization and constraint satisfaction for each generation in the evolution process by using a dynamic threshold $\varepsilon$ to balance the selection of feasible and infeasible solutions. This type of $\varepsilon$ constraint-handling method can effectively solve most complex CMOPs, especially by quickly finding the best feasible region in the CMOP. However, this method of mixed consideration and selection of feasible and infeasible solutions is not conducive to the feasible solutions converging well to the PF of the objective functions. The PSDS uses the $fr$ and the $PND$ to accurately determine the population’s state in the objective space of the CMOP to determine whether the next generation of evolution is to consider the constraints. After detecting the population state, whether to consider constraints is fed back to the environmental selection, and the positive impact on the evolutionary process is shown in Figure 4. Additionally, while testing the population state in each generation, we check the population periodically with $T$ as a cycle to determine if the population is in a dilemma that cannot continue to be optimized. Suppose it is determined in a certain inspection that the population has not gained evolutionary benefits compared with the state ten generations before [36]. In that case, the population will be re-initialized to provide more possibilities to search for all feasible regions. Among them, $T$ is a customized period parameter, which is set as $T = T_{max}/10$ in this paper. In combination with the algorithm, since only the PSDS is used in the first 70% of the evolutionary process, in the extreme case where the population is in a dilemma during each detection, the number of reinitializations of the population will reach the maximum of six times. However, in reality, during the running of the algorithm, this number of times will be less than six times. Figure 5 shows the positive effect of the population restart schema in solving CMOPs.

The unconstrained fitness value calculation is triggered in the environmental selection stage where the population converges to the unconstrained PF. If the unconstrained PF is close to the feasible regions, feasible solutions can be generated through the subsequent constraint fitness evaluation to pull the population back to the feasible region [31]. However, when the unconstrained PF is far from the feasible region, it may not be possible to generate feasible offspring through genetic operations. It causes the population to be trapped in infeasible space (Figure 5 (a)), and it is difficult to achieve the ultimate feasibility of the population. It is effective and necessary to reinitialize the population (Figure 5 (b)) to search for other feasible regions that may exist (Figure 5 (c)). Finally, the population can better converge in all feasible regions (Figure 5 (d)). The specific implementation methods and parameter settings of the population state detection and restart schema can be seen in Algorithm 3.

Before reinitializing the population, if there are feasible solutions in the population, the population is stagnant in the feasible region. Then the unconstrained fitness evaluation should be used after reinitialization to make the population jump out of the feasible area that has been searched. Otherwise, it means the population is trapped in the infeasible regions, and the constrained fitness assessment should be used after reinitialization to allow the population to converge in the feasible regions.
4. Experimental design and analysis

In this section, we report on a series of experiments based on the PlatEMO [12]. To verify the superiority of PSDS compared with other algorithms in solving CMOPs, PSDS is first compared to five representative CMOEAs — NSGAII-CDP [13], COMEAD [43], ToP [29], PPS [31], and C-TAEA [32] — on four challenging benchmark suites: MW [20], CDTLZ [19], LIROMOP [20], DASCMP [14]. Then, we analyze the experimental results and demonstrate the superiority of our proposed algorithm.

4.1. Compared Algorithms and Benchmark Suites

The five selected CMOEAs for comparison are characteristic and representative algorithms in solving CMOPs. NSGAII-CDP always adheres to the principle of the supremacy of feasible solutions so that solutions converge well within the feasible region. CMOEAD is based on a decomposition framework and uses a series of uniform weight vectors as the directions of population convergence so that the population has
excellent distribution. ToP and PPS use different optimization strategies in two stages in order to make the population cross infeasible regions in the early stage. C-TAEA uses a dual-population strategy, in which one population converges within the constrained regions, and the other population searches for all feasible regions in the objective space, allowing the two populations to exchange beneficial information. These four benchmark test suites contain many CMOPs. There are various feasible regions that are difficult to search in these CMOPs, including large infeasible regions, separated PFs, disconnected PFs, unevenly distributed PFs and extremely small feasible regions. These complex CMOPs bring stiff challenges for CMOEAs trying to find the optimal solutions.

4.2. General parameters

1) Genetic evolutionary manipulation: NSGAII-CDP, CMOEAD and C-TAEA use simulated binary crossover (SBX) \cite{38} and the polynomial mutation (PM) \cite{39}. The crossover probability in \text{SBX} is set to 1, and the mutation probability in \text{PM} is set to \(1/D\) (\(D\) is the number of decision variables), and the distribution index of crossover and mutation is set to 20. ToP and PPS use differential evolution (DE) \cite{45} and \text{PM}. The parameters \(CR\) and \(F\) in DE are set to 1 and 0.5, respectively. The PSDS used \text{SBX} and \text{PM} on MW and CDTLZ, then used the DE and \text{PM} on LIRCMOP and DASCMOP.

2) Parameter settings in each algorithm: During our comparison experiments, the parameter settings of other algorithms completely followed the parameter setting principles mentioned in their respective original articles. This was especially true for PPS, which is based on the MOEA/D framework. The parameter settings in PPS were as follows: \(\alpha = 0.95; \tau = 0.1; cp = 2\) and \(l = 20\). For our PSDS, \(\text{isCons} = \text{true}; T = T_{\text{max}}/10\).

3) Parameter settings in CMOPs: Different test problems have a different number of objective functions and various difficulty levels of trouble. This paper show the parameter settings of all test problems involved in Table 1. \(N\) denotes the population size; \(M\) represents the dimension of the objective space; \(D\) is the number of decision variables, and \(E\) is the number of evaluations we set to solve the corresponding problem.

4.3. Performance metrics

Since the PFs of MOPs can represent the optimal solution set, many performance metrics have been proposed to evaluate the convergence performance of MOEAs on MOPs. Among them, the most representative and convincing are the following two performance metrics:

1) Inverted Generational Distance (IGD) \cite{46}: The IGD metric simultaneously reflects the performance of convergence and diversity and is defined as follows:

\[
IGD(P, P^*) = \frac{\sum_{z \in P} \text{dis}(z, P^*)}{|P^*|},
\]

(7)

where \(P\) is a set of solutions obtained from the algorithm, and \(P^*\) is approximately 10,000 uniformly distributed points sampled on the PF. \(\text{dis}(z, P)\) is the Euclidean distance between \(z\) and its nearest neighbor in \(P\).
<table>
<thead>
<tr>
<th>Problem</th>
<th>N</th>
<th>M</th>
<th>D</th>
<th>E</th>
<th>Problem</th>
<th>N</th>
<th>M</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW1</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP1</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW2</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP2</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW3</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP3</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW4</td>
<td>105</td>
<td>3</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP4</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW5</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP5</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW6</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP6</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW7</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP7</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW8</td>
<td>105</td>
<td>3</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP8</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW9</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP9</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW10</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP10</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW11</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP11</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW12</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP12</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW13</td>
<td>100</td>
<td>2</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP13</td>
<td>105</td>
<td>3</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>MW14</td>
<td>105</td>
<td>3</td>
<td>15</td>
<td>100000</td>
<td>LIRCMOP14</td>
<td>105</td>
<td>3</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>C1-DTLZ1</td>
<td>105</td>
<td>3</td>
<td>7</td>
<td>300000</td>
<td>DASCMOP1</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>C1-DTLZ3</td>
<td>105</td>
<td>3</td>
<td>12</td>
<td>300000</td>
<td>DASCMOP2</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>C2-DTLZ2</td>
<td>105</td>
<td>3</td>
<td>12</td>
<td>300000</td>
<td>DASCMOP3</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>C3-DTLZ4</td>
<td>105</td>
<td>3</td>
<td>12</td>
<td>300000</td>
<td>DASCMOP4</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC1-DTLZ1</td>
<td>105</td>
<td>3</td>
<td>7</td>
<td>300000</td>
<td>DASCMOP5</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC1-DTLZ3</td>
<td>105</td>
<td>3</td>
<td>12</td>
<td>300000</td>
<td>DASCMOP6</td>
<td>100</td>
<td>2</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC2-DTLZ1</td>
<td>105</td>
<td>3</td>
<td>7</td>
<td>300000</td>
<td>DASCMOP7</td>
<td>105</td>
<td>3</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC2-DTLZ3</td>
<td>105</td>
<td>3</td>
<td>12</td>
<td>300000</td>
<td>DASCMOP8</td>
<td>105</td>
<td>3</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC3-DTLZ1</td>
<td>105</td>
<td>3</td>
<td>7</td>
<td>300000</td>
<td>DASCMOP9</td>
<td>105</td>
<td>3</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC3-DTLZ3</td>
<td>105</td>
<td>3</td>
<td>12</td>
<td>300000</td>
<td>DASCMOP10</td>
<td>105</td>
<td>5</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC1-DTLZ1</td>
<td>105</td>
<td>5</td>
<td>9</td>
<td>300000</td>
<td>DASCMOP11</td>
<td>105</td>
<td>8</td>
<td>10</td>
<td>300000</td>
</tr>
<tr>
<td>DC1-DTLZ3</td>
<td>105</td>
<td>5</td>
<td>14</td>
<td>300000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC2-DTLZ1</td>
<td>105</td>
<td>8</td>
<td>12</td>
<td>300000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC2-DTLZ3</td>
<td>105</td>
<td>8</td>
<td>17</td>
<td>300000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2) *Hypervolume (HV)*: The HV metric can provide convergence and distribution information of the algorithm simultaneously and has been accepted as the most commonly used indicator in comparisons. HV measures the volume of the hypervolume dominated by an approximation set. It can be expressed as:

\[
HV(S) = \text{VOL}\left(\bigcup_{i=1}^{n} [y_i(x_i), z_i'] \times \ldots \times [f_n(x), z_n']\right),
\]

where \(\text{VOL}(\cdot)\) is the Lebesgue measure; \(m\) denotes the number of objectives, and \(z' = (z'_1, \ldots, z'_m)\) is a user-defined reference point in the objective space.

4.4. Experimental Results on Benchmark Suites

**Comparisons on MW Suite**: MW is a very general test suite for CMOPs which covers a wide range of constraint problem types with different characteristics, as such as where some feasible regions are disconnected (MW5, 9, 11, 12, 13) and feasible regions do not have an evenly distributed PF in MW2. Table 2 lists the comparison of the IGD values and HV values between PSDS and the five comparison CMOEAs on MW1-MW14. As we can see in the table, PSDS had the best overall performance based on the IGD value because it got the best results on nine CMOPs and had competitive results on the other five CMOPs.

![Table 2: Comparison results on IGD metric and HV metric for PSDS and the other algorithms on MW benchmark suite.](image)

The best result for each row is highlighted. "NS/NA" means that no feasible solution can be found. "+", "-", and "=" indicate that the result is significantly better, significantly worse, and statistically similar to the results obtained by PSDS, respectively.
Figure 6: Scatter plots of the population obtained by NSGAII-CDP, CMOEAD, ToP, PPS, C-TAEA, and the PSDS on MW2 (median IGD value). The red line is the constrained PF; the gray area is the feasible region.

Figure 6 shows the distribution of the six algorithms on MW2. The feasible region of MW2 is unevenly distributed in part close to the PF, and some parts of the feasible region are extremely narrow when approaching the PF, which leads to an uneven distribution of points on the PF. In Figure 6, all CMOEAs can approach the constraint PF, but only the PSDS can fully converge to the constrained PF. This is because other algorithms optimize the feasibility of the population from start to finish, making the population unable to explore the extremely narrow feasible region close to the unconstrained PF. PSDS uses the unconstrained convergence mechanism after the population converges in the feasible region, allowing the solutions to be optimized, thereby searching for extremely narrow feasible regions. At the same time, in the process of multiple searches after reinitialization, the archive set is used to save all the optimal solutions found. Therefore, it has good performance on this complex problem.

Comparisons on CDTLZ Suite: The test suite of the CDTLZ series contains ten constrained multi-objective test problems, each of which involves three-dimensional objective function optimization by default. In this test suite, the feasible region with some test problems is severely divided by infeasible space as in C1-DTLZ3. The DC-DTLZ series test problems are CMOPs that support the test of more than three objective functions and have complex decision space constraints. Table 3 presents the IGD values and HV values of the six compared MOEAs on the fourteen CDTLZ problems. From the test results, it can be seen that PSDS is in a leading position in most of the default three-dimensional CMOPs, and only the two problems of C1-DTLZ1 and C3-DTLZ4 are weaker than CMOEAD. It is weaker than C-TAEA in high-dimensional CMOPs, but the results are still competitive, much stronger than the other four comparison algorithms except C-TAEA. There are two main reasons why PSDS is weaker than C-TAEA. Objectively speaking, the dual-population strategy proposed by C-TAEA is mainly designed to be
The best result for each row is highlighted. "NaN(NaN)" means that no feasible solution can be found. "+", "-" and "=“ indicates that the result is significantly better, significantly worse, and statistically similar to the results obtained by PSDS, respectively.

Table 3: Comparison results on IGD metric and HV metric for PSDS and the other algorithms on C-DTLZ benchmark suite.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Metric</th>
<th>NSGAII</th>
<th>CMOEAD</th>
<th>ToP</th>
<th>PPS</th>
<th>C-TAEA</th>
<th>PSDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC1_DTLZ1</td>
<td>IGD</td>
<td>7.9780e-1 (2.86e-3)</td>
<td>7.5413e-1 (3.48e-3)</td>
<td>8.4347e-1 (3.21e-3)</td>
<td>7.9780e-1 (2.86e-3)</td>
<td>7.5413e-1 (3.48e-3)</td>
<td>8.4347e-1 (3.21e-3)</td>
</tr>
<tr>
<td>DC1_DTLZ1</td>
<td>HV</td>
<td>6.6667e-1 (3.10e-2)</td>
<td>6.6667e-1 (3.10e-2)</td>
<td>6.6667e-1 (3.10e-2)</td>
<td>6.6667e-1 (3.10e-2)</td>
<td>6.6667e-1 (3.10e-2)</td>
<td>6.6667e-1 (3.10e-2)</td>
</tr>
<tr>
<td>DC2_DTLZ1</td>
<td>IGD</td>
<td>4.4897e-1 (1.34e-3)</td>
<td>4.4897e-1 (1.34e-3)</td>
<td>4.4897e-1 (1.34e-3)</td>
<td>4.4897e-1 (1.34e-3)</td>
<td>4.4897e-1 (1.34e-3)</td>
<td>4.4897e-1 (1.34e-3)</td>
</tr>
<tr>
<td>DC2_DTLZ1</td>
<td>HV</td>
<td>5.3671e-1 (3.59e-2)</td>
<td>5.3671e-1 (3.59e-2)</td>
<td>5.3671e-1 (3.59e-2)</td>
<td>5.3671e-1 (3.59e-2)</td>
<td>5.3671e-1 (3.59e-2)</td>
<td>5.3671e-1 (3.59e-2)</td>
</tr>
</tbody>
</table>

Figure 7: Scatter plots of the population obtained by NSGAII-CDP, CMOEAD, ToP, PPS, C-TAEA and the PSDS on DC2-DTLZ3 (median IGD value). The gray area is the Pareto Front.

20
suitable for solving the problem of high-dimensional CMOPs, and C-TAEA is also the author of the DCDTLZ series of problems. Subjectively speaking, the PSDS in this paper involves a restart schema. In the face of slow convergence of high-dimensional CMOPs, reinitializing the population multiple times may interrupt the unfinished convergence process, thereby affecting the algorithm’s convergence.

In Section 2, Figure 3 shows the experimental effects of some of the algorithms on the C1-DTLZ3 test problem. The infeasible area of DC2-DTLZ3 is extremely difficult to cross. The running results of the six algorithms on this test problem are shown in Figure 4. The three algorithms NSGAII-CDP, CMOEAD, and ToP cannot cross the infeasible area, and even the ToP cannot find any feasible solutions. Although the other two algorithms PPS and C-TAEA do not become trapped in the local optimum, they cannot converge well on the constrained PF. This is mainly reflected in the poor convergence of PPS in the feasible region. In addition, although C-TAEA has converged to the true PF, mainly due to the substantial convergence ability after ignoring constraints and having good distribution based on uniform weight vector selection.

Table 4: Comparison results on IGD metric and HV metric for PSDS and the other algorithms on LIR-CMOP benchmark suite.

<table>
<thead>
<tr>
<th>Problem</th>
<th>PS</th>
<th>Metric</th>
<th>NSGAII</th>
<th>CMOEAD</th>
<th>Tp</th>
<th>PPS</th>
<th>C-TAEA</th>
<th>PSDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIRCMOP 1</td>
<td>2</td>
<td>IGD</td>
<td>2.1500e-0 (2.10e-2)</td>
<td>2.0569e-0 (2.15e-2)</td>
<td>1.6606e-1 (3.46e-2)</td>
<td>1.6906e-1 (3.22e-1)</td>
<td>1.6799e-1 (3.10e-2)</td>
<td>1.6312e-1 (2.39e-2)</td>
</tr>
<tr>
<td>LIRCMOP 2</td>
<td>2</td>
<td>IGD</td>
<td>1.9597e-1 (3.57e-2)</td>
<td>1.9583e-1 (3.56e-2)</td>
<td>1.1360e-1 (1.96e-1)</td>
<td>0.0000e+0 (0.00e+0)</td>
<td>1.1339e-1 (2.86e-2)</td>
<td>2.9182e-1 (5.30e-1)</td>
</tr>
<tr>
<td>LIRCMOP 3</td>
<td>2</td>
<td>IGD</td>
<td>2.3062e-1 (1.19e-2)</td>
<td>2.3085e-1 (1.20e-2)</td>
<td>2.0806e-1 (3.17e-2)</td>
<td>2.0643e-1 (1.11e-1)</td>
<td>1.3739e-1 (1.23e-2)</td>
<td>2.0382e-1 (3.60e-2)</td>
</tr>
<tr>
<td>LIRCMOP 4</td>
<td>2</td>
<td>IGD</td>
<td>2.8045e-1 (1.20e-2)</td>
<td>2.6709e-1 (1.48e-2)</td>
<td>1.2596e-1 (1.96e-2)</td>
<td>1.3912e-1 (2.96e-2)</td>
<td>1.2755e-1 (1.34e-1)</td>
<td>2.5672e-1 (3.96e-2)</td>
</tr>
<tr>
<td>LIRCMOP 5</td>
<td>2</td>
<td>IGD</td>
<td>1.0601e-1 (1.10e-2)</td>
<td>1.0400e-1 (1.10e-2)</td>
<td>5.7450e-2 (3.76e-3)</td>
<td>2.1056e-1 (1.93e-3)</td>
<td>1.8215e-1 (7.18e-2)</td>
<td>2.2518e-1 (3.10e-2)</td>
</tr>
<tr>
<td>LIRCMOP 6</td>
<td>2</td>
<td>IGD</td>
<td>2.1089e-1 (1.14e-2)</td>
<td>2.1178e-1 (1.60e-2)</td>
<td>1.8416e-1 (1.60e-2)</td>
<td>3.0162e-1 (2.57e-2)</td>
<td>2.3215e-1 (4.48e-2)</td>
<td>3.1329e-1 (4.09e-2)</td>
</tr>
<tr>
<td>LIRCMOP 7</td>
<td>2</td>
<td>IGD</td>
<td>3.0843e-1 (4.29e-2)</td>
<td>2.7386e-1 (4.41e-2)</td>
<td>3.3594e-1 (1.99e-2)</td>
<td>1.7113e-2 (3.36e-2)</td>
<td>2.1750e-1 (1.33e-1)</td>
<td>1.2792e-1 (2.35e-1)</td>
</tr>
<tr>
<td>LIRCMOP 8</td>
<td>2</td>
<td>IGD</td>
<td>2.3925e-1 (1.79e-2)</td>
<td>2.0939e-1 (2.93e-2)</td>
<td>2.6826e-1 (2.36e-2)</td>
<td>1.4133e-1 (1.84e-2)</td>
<td>1.4133e-1 (1.84e-2)</td>
<td>1.4133e-1 (1.84e-2)</td>
</tr>
</tbody>
</table>

Comparisons on LIR-CMOP Suite: The problems of the LIR-CMOP suite have become a huge challenge for most CMOEAs because their feasible regions are extremely narrow and close to the curve (LIRCMOP1–LIRCMOP8). Even feasible areas are divided into multiple small blocks by large infeasible space (LIRCMOP9–LIRCMOP12). As can be seen from the experimental results of the IGD values and HV values of the six compared MOEAs on the fourteen LIR-CMOP problems in Table 4, the performance of the PSDS
on 11 problems was better than the other algorithms. In addition, on another problem (LIRCMOP2), the PSDS was comparable to the best performing PPS. The remaining two problems (LIRCMOP1 and LIRCMOP5) could only be solved better by the PPS algorithm. In contrast, none of the other four algorithms achieved the best performance on this suite of test problems.

Figure 8 plots the population of the PSDS and other comparison algorithms on the LIR-CMOP10 problem, whose PFs are disconnected by discontinuous feasible regions and at the same time obstructed by a large infeasible region in the direction of convergence. Such an extremely difficult CMOP causes NSGAII-CDP and CMOEAD to be unable to approach the PF. ToP, and C-TAEA cannot find all the PFs even if it crosses the infeasible area. Even if ToP and C-TAEA cross the infeasible area, they cannot find all the smaller feasible regions and have a good distribution in the entire PF. Only PPS and the PSDS can converge to all feasible regions, and the stability of the PSDS is better to ensure that each solution can converge to the complete PF.

Comparisons on DAS-CMOP Suite: DAS-CMOP is a benchmark suite with characteristics such as having a type and difficulty of constraint problem that can be customized by the user. The feasible regions of most CMOPs in this test suite are separated from each other and are far away from the unconstrained PF. Our parameter settings for each CMOP in the DAS benchmark suite followed the default parameters in the experimental platform PlatEMO [42]. In addition, it also uses the characteristics of the DAS benchmark test to independently design DASCMOP10 with five objective functions and DASCMOP11 with eight objective functions based on the definition of the CMOPs of the scalable number of objectives in the original paper of DASCMOP [44]. It can be seen from the experimental results of the six algorithms in Table 5 that
the PSDS reflects promising performance because it achieved the best testing results on five problems, including DAS-CMOP1, DAS-CMOP2, DAS-CMOP3, DAS-CMOP6 and DAS-CMOP9. This is most likely because most of them have narrow and easily overlooked feasible regions. The PSDS had certain weaknesses on DAS-CMOP4, DAS-CMOP5, DAS-CMOP7 and DAS-CMOP8 since their requirements for the diversity of solutions are extremely high. In the experimental results of the other two newly constructed constrained many-objective optimization problems, DAS-CMOP10 and DAS-CMOP11, the IGD value of PSDS is the best, but the HV value is slightly weaker than the other two algorithms. The reason is that when PSDS solves high-dimensional problems, the convergence and distribution of the algorithm cannot achieve the best results at the same time.

Table 5: Comparison results on IGD metric and HV metric for PSDS and the other algorithms on DAS-CMOP benchmark suite.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Metric</th>
<th>NSGAII</th>
<th>CoMOEAD</th>
<th>ToP</th>
<th>PPS</th>
<th>CTAA</th>
<th>PSDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAS-CMOP1</td>
<td>IGD</td>
<td>7.0685e-1</td>
<td>6.9134e-1</td>
<td>4.9075e-1</td>
<td>4.12e-2</td>
<td>1.5195e-1</td>
<td>1.806e-1</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>7.141e-1</td>
<td>5.75e-2</td>
<td>3.51e-2</td>
<td>1.57e-1</td>
<td>1.707e-1</td>
<td>1.35e-3</td>
</tr>
<tr>
<td>DAS-CMOP2</td>
<td>IGD</td>
<td>2.57e-2</td>
<td>2.10e-2</td>
<td>4.16e-3</td>
<td>1.51e-3</td>
<td>5.17e-3</td>
<td>8.6e-3</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>3.21e-1</td>
<td>1.21e-2</td>
<td>1.42e-1</td>
<td>1.09e-1</td>
<td>5.19e-1</td>
<td>1.8e-2</td>
</tr>
<tr>
<td>DAS-CMOP3</td>
<td>IGD</td>
<td>9.10e-1</td>
<td>1.96e-1</td>
<td>6.68e-2</td>
<td>2.23e-1</td>
<td>1.34e-1</td>
<td>1.2e-3</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>2.06e-1</td>
<td>1.19e-1</td>
<td>2.11e-1</td>
<td>5.6e-2</td>
<td>2.23e-1</td>
<td>1.32e-3</td>
</tr>
<tr>
<td>DAS-CMOP4</td>
<td>IGD</td>
<td>1.09e1</td>
<td>1.24e-2</td>
<td>2.11e-1</td>
<td>NaN</td>
<td>NaN</td>
<td>3.13e-1</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>2.41e-1</td>
<td>1.44e-1</td>
<td>1.79e-1</td>
<td>1.7e-2</td>
<td>1.68e-1</td>
<td>1.53e-3</td>
</tr>
<tr>
<td>DAS-CMOP5</td>
<td>IGD</td>
<td>5.65e-3</td>
<td>8.15e-1</td>
<td>1.82e-2</td>
<td>NaN</td>
<td>NaN</td>
<td>4.22e-3</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>3.5e-1</td>
<td>1.49e-4</td>
<td>1.97e-1</td>
<td>1.7e-2</td>
<td>1.7e-1</td>
<td>2.25e-3</td>
</tr>
<tr>
<td>DAS-CMOP6</td>
<td>IGD</td>
<td>3.3e-1</td>
<td>6.01e-2</td>
<td>1.18e0</td>
<td>2.32e-1</td>
<td>2.2e-3</td>
<td>2.22e-2</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>1.34e-1</td>
<td>2.87e-1</td>
<td>0.00e+0</td>
<td>2.17e-1</td>
<td>2.1e-4</td>
<td>3.11e-3</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>2.83e-1</td>
<td>2.88e-1</td>
<td>NaN</td>
<td>2.81e-1</td>
<td>1.77e-3</td>
<td>2.88e-1</td>
</tr>
<tr>
<td>DAS-CMOP8</td>
<td>IGD</td>
<td>6.07e-2</td>
<td>5.52e-2</td>
<td>NaN</td>
<td>6.43e-2</td>
<td>5.47e-2</td>
<td>2.08e-3</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>2.05e-1</td>
<td>2.06e-1</td>
<td>NaN</td>
<td>2.01e-1</td>
<td>1.54e-3</td>
<td>2.04e-1</td>
</tr>
<tr>
<td>DAS-CMOP9</td>
<td>IGD</td>
<td>3.68e-1</td>
<td>2.79e-1</td>
<td>5.71e-1</td>
<td>9.35e-2</td>
<td>1.66e-1</td>
<td>6.05e-2</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>1.30e-1</td>
<td>1.50e-1</td>
<td>8.78e-2</td>
<td>1.90e-1</td>
<td>1.67e-1</td>
<td>3.09e-1</td>
</tr>
<tr>
<td>DAS-CMOP10</td>
<td>IGD</td>
<td>3.2e-1</td>
<td>2.87e-1</td>
<td>5.26e-1</td>
<td>3.16e-1</td>
<td>3.49e-1</td>
<td>3.11e-1</td>
</tr>
<tr>
<td></td>
<td>HV</td>
<td>1.82e-2</td>
<td>1.98e-3</td>
<td>0.00e+0</td>
<td>9.67e-4</td>
<td>1.97e-3</td>
<td>3.59e-3</td>
</tr>
</tbody>
</table>

The best result for each row is highlighted. "NaN(NaN)" means that no feasible solution can be found. "+", "-" and "=" indicate that the result is significantly better, significantly worse, and statistically similar to the results obtained by PSDS, respectively.

The population distribution of the PSDS and the other five algorithms on DAS-CMOP3 is shown in Figure 9. In such a problem, there are a large number of feasible regions, and some feasible regions are even extremely narrow. Although the PSDS can search as many PFs as possible compared to other algorithms, it cannot find all nondominated solutions like other algorithms can. In the DAS-CMOP3 of this experiment, there are two solutions in the optimal solution set. One of them is easily misunderstood as being dominated by other solutions, and the feasible region of the other is almost the point itself, so these two points are very difficult to search. There is another type of problem in the DAS test suite, represented by DAS-CMOP9, with many separate feasible regions.
Figure 9: Scatter plots of the population obtained by NSGAII-CDP, CMOEAD, ToP, PPS, C-TAEA and the PSDS on DAS-CMOP3 (median IGD value). The red points are the constrained PF; the gray area is the feasible region.

4.5. Effectiveness of state detection strategy of PSDS

This subsection uses ablation research to compare several types of algorithms that only use partial improvements. Using this partial variable method for experimental comparison can verify the effectiveness of the core components of the proposed PSDS. We conducted an experimental comparison between PSDS and some of its variants on the MW benchmark test suite. Among them, the strategy of only retaining the restart schema must be effective only in the evolution process after the last initialization. Therefore, this single strategy, which is obviously uncompetitive, will not be compared with PSDS. The first variant PSDS\(^1\) only discards the population state detection strategy we proposed and retains the restart schema and the archive set that can provide a stability guarantee for the restart schema. This PSDS\(^1\) verifies whether the main state detection strategy in this paper plays a decisive role in the algorithm’s performance. The second variant PSDS\(^2\) retains the state detection strategy. To compare with the main algorithm, we removed the archive set of excellent solutions, allowing the population to be reinitialized at the appropriate situation after state detection. The third variant PSDS\(^3\) is also a retained state detection strategy. The difference from PSDS\(^2\) is that PSDS\(^3\) keeps the archive set of elite solutions and no longer considers reinitializing the population. These two variants PSDS\(^2\) and PSDS\(^3\) can thoroughly verify the benefits of the other two auxiliary strategies on the basis of retaining the state detection strategy. It is worth mentioning that state detection only provides reference information for adjusting the evolution process, so there is no design of algorithm variant that only includes the state detection strategy.

Table 6 presents the IGD results of PSDS and the other three PSDS variants on MW1-MW14. It is evident from the results that the overall performance of the full version of PSDS is better than other variants. Especially from the comparison with
Table 6: Comparison results of IGD metric obtained by PSDS and its three variants on the MW benchmark suite. Best result in each row is highlighted.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Metric</th>
<th>PSDS¹</th>
<th>PSDS²</th>
<th>PSDS³</th>
<th>PSDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW1</td>
<td>IGD</td>
<td>4.189e-3 (2.78e-3) -</td>
<td>3.990e-2 (8.41e-2) -</td>
<td>1.629e-3 (3.03e-5) =</td>
<td>1.809e-3 (1.01e-3)</td>
</tr>
<tr>
<td>MW2</td>
<td>IGD</td>
<td>6.504e-3 (3.75e-3) =</td>
<td>2.335e-2 (1.66e-2) -</td>
<td>3.240e-2 (2.02e-2) -</td>
<td>5.402e-3 (2.98e-3)</td>
</tr>
<tr>
<td>MW3</td>
<td>IGD</td>
<td>5.186e-3 (2.97e-4) -</td>
<td>6.566e-3 (6.04e-4) -</td>
<td>4.856e-3 (2.45e-4) +</td>
<td>5.038e-3 (2.78e-4)</td>
</tr>
<tr>
<td>MW4</td>
<td>IGD</td>
<td>3.867e-2 (4.36e-4) -</td>
<td>4.513e-2 (3.25e-2) -</td>
<td>3.809e-2 (6.18e-3) =</td>
<td>3.831e-2 (1.65e-4)</td>
</tr>
<tr>
<td>MW5</td>
<td>IGD</td>
<td>2.940e-3 (5.35e-3) =</td>
<td>1.172e-1 (1.97e-1) -</td>
<td>2.308e-1 (2.24e-3) -</td>
<td>1.014e-3 (7.97e-4)</td>
</tr>
<tr>
<td>MW6</td>
<td>IGD</td>
<td>6.158e-3 (4.87e-3) -</td>
<td>5.239e-2 (1.13e-1) -</td>
<td>2.253e-2 (1.59e-2) -</td>
<td>3.962e-3 (2.56e-3)</td>
</tr>
<tr>
<td>MW7</td>
<td>IGD</td>
<td>4.574e-3 (2.20e-4) +</td>
<td>3.367e-2 (2.32e-2) -</td>
<td>4.672e-3 (4.79e-4) +</td>
<td>4.844e-3 (3.10e-4)</td>
</tr>
<tr>
<td>MW8</td>
<td>IGD</td>
<td>4.508e-2 (2.64e-3) =</td>
<td>5.429e-2 (2.01e-2) -</td>
<td>4.845e-2 (4.33e-3) -</td>
<td>4.458e-2 (4.92e-4)</td>
</tr>
<tr>
<td>MW9</td>
<td>IGD</td>
<td>3.669e-2 (1.19e-1) -</td>
<td>1.031e-1 (2.42e-1) -</td>
<td>5.000e-2 (1.41e-1) =</td>
<td>4.647e-3 (1.06e-3)</td>
</tr>
<tr>
<td>MW10</td>
<td>IGD</td>
<td>4.740e-3 (2.35e-3) -</td>
<td>7.019e-2 (1.11e-1) -</td>
<td>2.009e-1 (1.64e-1) -</td>
<td>3.821e-3 (1.38e-3)</td>
</tr>
<tr>
<td>MW11</td>
<td>IGD</td>
<td>3.922e-1 (3.38e-1) -</td>
<td>7.972e-2 (1.33e-1) -</td>
<td>6.267e-3 (3.07e-4) =</td>
<td>6.352e-3 (2.79e-4)</td>
</tr>
<tr>
<td>MW12</td>
<td>IGD</td>
<td>4.714e-3 (1.50e-4) -</td>
<td>9.136e-2 (2.31e-1) -</td>
<td>4.610e-3 (1.14e-4) -</td>
<td>4.531e-3 (1.01e-4)</td>
</tr>
<tr>
<td>MW13</td>
<td>IGD</td>
<td>3.357e-2 (8.58e-3) -</td>
<td>9.380e-2 (3.95e-2) -</td>
<td>1.071e-1 (7.42e-2) -</td>
<td>1.181e-2 (3.07e-3)</td>
</tr>
<tr>
<td>MW14</td>
<td>IGD</td>
<td>1.233e-1 (9.70e-3) =</td>
<td>1.531e-1 (1.23e-1) -</td>
<td>1.173e-1 (3.56e-3) =</td>
<td>1.189e-1 (3.51e-3)</td>
</tr>
</tbody>
</table>

The best result for each row is highlighted. "+", "-" and "=" indicate that the result is significantly better, significantly worse, and statistically similar to the results obtained by PSDS, respectively. PSDS¹ removes the population state detection strategy on the basis of PSDS; PSDS² removes the archive set of feasible nondominated solutions on the basis of PSDS; PSDS³ removes the restart schema on the basis of PSDS.
PSDS$^1$, the population state detection strategy proposed in this paper plays a decisive role in improving algorithm performance. The experimental results show that PSDS$^2$ lags behind PSDS, which also means that the auxiliary mechanism of restart schema is not a panacea. The effectiveness of the restart schema can only play a better role under the overall framework of PSDS. Only PSDS$^3$ is ahead of PSDS on a small number of CMOPs such as MW1, MW3, MW4, MW11. The reason is that these CMOPs have common characteristics, that is, these problems are not complex and all only test the convergence of the algorithm in the feasible region. Compared with PSDS, PSDS$^3$ only removes the restart schema. Although it has better convergence in these simple problems, there is no apparent gap between PSDS and PSDS$^3$.

On the contrary, PSDS$^3$ has a considerable gap with PSDS in complex CMOPs that focus on testing distribution. Through the experimental analysis in this section, it can be summarized that PSDS is a multi-strategy collaborative optimization stability algorithm. Every core component has its necessity to be considered. In addition, the restart schema in PSDS partially weakens the convergence performance of the algorithm, which is a dilemma. Still, it cannot deny the restart schema’s contribution to ensuring the diversity and stability of the algorithm.

5. Conclusions

In this paper, we have proposed a PSDS for CMOEAs to solve CMOPs. The main idea of the proposed algorithm is to detect the specific state of the population in each iteration process, then select an appropriate environmental selection strategy based on the state to allow the population to continue to obtain evolutionary benefits. On this basis, the population restart scheme continuously improves the previous solutions. Then the archive set can complement the advantages of the detection strategy and restart mechanism to achieve a stable performance of the entire algorithm.

Experimental results show that the proposed algorithm had better overall performance than the other five algorithms on a total of 53 test problems in four test suites. This demonstrates that the proposed strategy has a strong convergence ability on problems with large infeasible regions and has strong search ability in problems with discrete and small feasible regions.

By summarizing the difficulties of current CMOPs and analyzing the deficiencies of existing CMOEAs in solving these difficult problems, this work has shown the importance of balancing objective optimization and constraint satisfaction through reasonable adjustment of environmental selection strategies when solving CMOPs. It is desirable to extend the proposed detection strategy and adopt more effective environmental selection strategies for solving other more challenging CMOPs.

6. Acknowledgements

The authors wish to thank the support of the National Natural Science Foundation of China (Grant No. 61876164, 61772178), The MOEA Key Laboratory of Intelligent Computing and Information Processing, the Science and Technology Plan Project of Hunan Province (Grant No. 2016TP1020), the Natural Science Foundation of Hunan Province (Grant No. 2020JJ4590), the Provinces and Cities Joint Foundation Project (Grant No. 26...
References


A Constrained Multi-objective Evolutionary Strategy Based on Population State Detection
— Supplemental Material

Huanrong Tang\textsuperscript{a,b}, Fan Yu\textsuperscript{a,b}, Juan Zou\textsuperscript{a,b}, Shengxiang Yang\textsuperscript{a,d}, Jinhua Zheng\textsuperscript{a,b,c}

\textsuperscript{a}Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Computer science College of Xiangtan University, Xiangtan, Hunan Province, China
\textsuperscript{b}School of Computer Science and School of Cyberspace Science University of Xiangtan, Xiangtan, 411105, China
\textsuperscript{c}Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China
\textsuperscript{d}School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.


![Figure 1: Scatter plots of the population obtained by NSGAII-CDP, CMOEAD, ToP, PPS, C-TAEA and the PSDS on MW8 (median IGD value). The gray area is the Pareto Front.](image)

Figure 1 plots the populations obtained by the six CMOEAs on MW8. We can see from this figure that MW8 is a three-objective function test problem, and the real feasible regions are separated. Almost all algorithms in the comparison experiment can find these feasible regions, but only the convergence and distribution of the PSDS in these feasible regions is the best. The best distribution is due to the uniform weight vector we adopted in solving the three-dimensional test problems.
In the C3-DTLZ4 test problem shown in Figure 2, the distribution of the PSDS on the constrained PF is only slightly weaker than CMOEAD, which is because there is no infeasible area to limit the convergence of the CMOEAD algorithm on this test problem.

Another type of LIRC MOP has multiple small feasible areas, and even the PFs in...
multiple feasible areas are composed of points (LIR-CMOP12). It can be seen from Figure 3 that only the PSDS can stably find all PFs when solving LIR-CMOP12. The fact that the PSDS could obtain such a stable search effect on LIR-CMOP12 is entirely due to the archive set A that preserves the existing beneficial solutions and the scheme that reinitializes the population.

![Scatter plots of the population obtained by NSGAII-CDP, CMOEAD, ToP, PPS, C-TAEA and the PSDS on DAS-CMOP3 (median IGD value).](image)

Figure 4: Scatter plots of the population obtained by NSGAII-CDP, CMOEAD, ToP, PPS, C-TAEA and the PSDS on DAS-CMOP3 (median IGD value). The red points are the constrained PF; the gray area is the feasible region.

The distribution of the solutions of the six algorithms on DAS-CMOP9 are shown in Figure 4. Due to the limited number of solutions in the population and a large number of separated feasible regions, it is difficult to distribute the solutions obtained by the algorithm evenly on all PFs. The population results show that the PSDS has better distribution performance to find as many sub-feasible regions as possible and make the solution converge on most PFs. In contrast, the distribution of other algorithms on this problem is not so perfect. The main reason is that the PSDS can use the convergence process after multiple reinitializations to find other sub-feasible regions that have not been searched.